A Simple Rule for Improving Solution Times in QAP Sparse Instances

G. Miranda [a,*]

[a] *Universidade Federal de Minas Gerais, Departamento de Engenharia de Produção, Av. Antônio Carlos, 6627, Pavilhão Central de Aulas - Pampulha - zip 30161-010 - Belo Horizonte - Minas Gerais - Brazil*

R. S. de Camargo [b]

[b] *Universidade Federal de Minas Gerais, Departamento de Ciência da Computação, Av. Antônio Carlos, 6627, Prédio do Icex, sala 4010 - Pampulha - zip 31270-010 - Belo Horizonte - Minas Gerais - Brazil*

H. P. Luna [c]

[c] *Universidade Federal de Alagoas, Instituto de Computação, Campus A. C. Simões, BR 104, Km 14 - Tabuleiro dos Martins - zip 57072-970 - Maceió - Alagoas - Brazil*

Abstract

The Quadratic Assignment Problem (QAP) is a well known NP-hard combinatiorial optimization problem. Instances of size greater than 30 are still a computational challenge, being even hard to solve when dealing with sparse demand matrices. Finding short formulations or ways of reducing the number of variables is a key matter for achieving success. In this note, we present a simple rule for reducing the number of variables of the Adams and Johnson (RLT level-1) formulation when addressing instances with sparse matrices. The efficiency of the proposed approach can be observed by the expressive reduction of the solution times.

Key words: Quadratic Assignment Problem, Compact Flow Formulation, RLT level-1 linearization

* Corresponding author.
Email addresses: miranda@dep.ufmg.br (G. Miranda), rcamargo@dcc.ufmg.br (R. S. de Camargo), pacca@tci.ufal.br (H. P. Luna).

1 Introduction

The pure (homogeneous) Quadratic Assignment Problem (QAP) assigns facilities to locations minimizing the total transportation cost of intermediate commodities. The QAP is a well known NP-hard combinatorial optimization problem, where solving instance of size greater than 30 remains a computational challenge. Computer times are even worse when sparse instances are considered.

Since the pioneering work of Koopmans and Beckmann [5] in 1957, many authors have been striven for improving bounds. Among them, the work of Adams and Johnson [2] is reputed to have the dominating linearization for this problem, excepting the formulation of Ramakrishnan *et al* [6]. However, the computational cost of obtaining these bounds is prohibitive.

Recently, Hahn *et al* [4] have demonstrated that the Adams and Johnson linearization is equivalent to the formulation of the QAP using the level-1 reformulation-linearization technique (RLT) proposed by Adams and Sherali [1]. They have also established the equivalence of Ramakrishnan *et al* [6] formulation and the RLT level-2 linearization for QAP. Their formulation has fewer variables and constraints than the traditional one.

In this note, we present a simple rule for reducing even further the number of variables and constraints of the Adams and Johnson formulation when facing sparse instances. The computational solution time is quite sensitive to the flow matrix sparsity degree.

2 QAP Formulations

The model of the QAP is stated for a given the set K of n facilities, a set I of n locations, and two n^2 matrices representing, respectively, costs between location pairs and demands between facility pairs. We also define intermediate commodities transportation costs as $b_{kl}c_{ij}x_{ki}x_{lj}$ $\forall k,l \in K$, $\forall i,j \in I$, where b_{kl} is the demand between the facility pair kl, c_{ij} is the transportation cost between the location pair ij, and x_{ki} (x_{lj}) is the integer variable stating the assignment of facility k (l) at location i (j). The x variables must constitute an assignment, implying the following formulation:

$$\text{Min} \sum_{k \in K} \sum_{i \in I} \sum_{l \in K} \sum_{j \in I} b_{kl} c_{ij} x_{ki} x_{lj} \tag{1}$$

$$\text{s. t.:} \tag{2}$$

$$\sum_{k \in K} x_{ki} = 1 \qquad \forall\, i \in I \tag{3}$$

$$\sum_{i \in I} x_{ki} = 1 \qquad \forall\, k \in K \tag{4}$$

$$x_{ki} \in \{0, 1\} \qquad \forall\, k \in K,\ i \in I \tag{5}$$

In 1994, Adams and Johnson derived a linearization for problem (1)-(5) that is reputed to dominate the other MILP relaxations of QAP. The Adams and Johnson formulation is stated as:

$$\text{Min} \sum_{k \in K} \sum_{i \in I} \sum_{l \in K} \sum_{j \in I} b_{kl} c_{ij} y_{kilj} \tag{6}$$

s. t.:

constraints (3) - (5)

$$\sum_{j \in I} y_{kilj} = x_{ki} \qquad \forall\, k, l \in K,\ i \in I,\ i \neq j,\ k \neq l \tag{7}$$

$$\sum_{l \in K} y_{kilj} = x_{ki} \qquad \forall\, k \in K,\ i, j \in I,\ i \neq j,\ k \neq l \tag{8}$$

$$y_{kilj} = y_{ljki} \qquad \forall\, k, l \in K,\ i, j \in I,\ i \neq j,\ k < l \tag{9}$$

$$y_{kilj} \geq 0 \qquad \forall\, k, l \in K,\ i, j \in I,\ i \neq j,\ k \neq l \tag{10}$$

Recently, Peter Hahn *et al* [4] have demonstrated how the formulation (6)-(10) is equivalent to a level-1 RLT linearization for QAP. They also have shown how to eliminate constraints (9) yielding:

$$\text{Min} \sum_{k\in K}\sum_{i\in I}\sum_{l\in K}\sum_{j\in I|i\neq j} b_{kl}c_{ij}y_{kilj} \tag{11}$$

s. t.:

constraints (3) - (5)

$$\sum_{j\in I|j\neq i} y_{kilj} = x_{ki} \qquad \forall\ k,l\in K, k<l,\ i\in I \tag{12}$$

$$\sum_{j\in I|j\neq i} y_{ljki} = x_{ki} \qquad \forall\ k,l\in K, k>l,\ i\in I \tag{13}$$

$$\sum_{l\in K|k<l} y_{kilj} + \sum_{l\in K|k>l} y_{ljki} = x_{ki} \quad \forall\ k\in K,\ i,j\in I,\ i\neq j \tag{14}$$

$$y_{kilj} \geq 0 \qquad \forall\ k,l\in K,\ k<l\ i,j\in I,\ i\neq j \tag{15}$$

The formulation (11)-(15) has less $n^2 \times \frac{(n-1)^2}{2}$ variables and constraints than the original one, (6)-(10). One can see that the product of assignment variables y_{kilj} can be reinterpreted as a flow from facility k to l through arc ij, written as f_{klij}. This simple maneuver implies the following formulation:

$$\text{Min} \sum_{(i,j)\in I^2|i\neq j}\ \sum_{(k,l)\in K^2|k<l} [c_{ij}b_{kl} + c_{ji}b_{lk}]f_{klij} \tag{16}$$

s. t.:

constraints $(3) - (5)$

$$\sum_{j\in I|j\neq i} f_{klij} = x_{ki} \qquad \forall\ i\in I,\ \forall k,l\in\ K, k<l \tag{17}$$

$$\sum_{i\in I|i\neq j} f_{klij} = x_{lj} \qquad \forall\ j\in I,\ \forall k,l\in\ K, k<l \tag{18}$$

$$\sum_{l\in K|k<l} f_{klij} + \sum_{l\in K|k>l} f_{lkji} = x_{ki} \qquad \forall\ k\in K,\ \forall i,j\in I, i\neq j \tag{19}$$

$$f_{klij} \geq 0 \qquad \forall\ i,j\in I, i\neq j,\ \forall k,l\in K,\ k<l \tag{20}$$

Constraints (17) can be seen as commodity flows that leave location i, while constraints (18) can be seen as commodity flows that arrive at location j, for a given kl pair. Constraints (19) assure that if facility k is placed at location i, for each arc (i,j) or (j,i) only one facility l is attended on this link.

When facing a sparse demand matrix, the formulation (16)-(20) has a degenerate objective function, once it has unnecessary flow variables. Anstreicher shows in [3] how the sparsity of an instance can be a major drawback for some bounding schemes. Furthermore, most solvers, including the leading edge Ilog

CPLEX, are specially sensitive to the number of variables in a given formulation. Finding short formulations or ways of reducing the number of variables is a key matter for achieving success. So the flow interpretation results on the following rule.

3 The Flow Elimination Rule

In order to discard the flow variables, we define the set:

$$\phi = \{(k,l) \in K^2 | k < l \;\text{ and }\; (b_{kl} \neq 0 \;\text{ or }\; b_{lk} \neq 0)\}$$

Therefore, creating flow variables only in this scope. The definition of the set ϕ avoids the creation of flow variables for kl pairs when both demand coefficients are zero. So, the QAP for sparse demand matrix can now be formulated as:

$$\text{Min} \sum_{(i,j)\in I^2|i\neq j} \; \sum_{(k,l)\in\phi} [c_{ij}b_{kl} + c_{ji}b_{lk}] f_{klij} \tag{21}$$

s. t.:

constraints $(3)-(5)$

$$\sum_{j\in I|j\neq i} f_{klij} = x_{ki} \qquad \forall\, i \in I,\; \forall (k,l) \in \;\phi \tag{22}$$

$$\sum_{i\in I|i\neq j} f_{klij} = x_{lj} \qquad \forall\, j \in I,\; \forall (k,l) \in \;\phi \tag{23}$$

$$\sum_{l\in\phi|k<l} f_{klij} + \sum_{l\in\phi|k>l} f_{lkji} \leq x_{ki} \qquad \forall\, k \in K,\; \forall i,j \in I, i \neq j \tag{24}$$

$$f_{klij} \geq 0 \qquad \forall\, i,j \in I, i \neq j,\; \forall (k,l) \in \phi \tag{25}$$

Constraints (24) are a relaxed version of constraints (19). This is due to the following fact: if we have no flow between facilities k and l placed at locations i and j respectively, the equality does not hold anymore. Hence, the formulation (21)-(25) is very compact and it does not have a degenerate objective function. The efficiency of the proposed approach can be observed by the expressive reduction of the solution times, as presented in section 4.

4 Computational Experience

We present on table 1 the results of the computational experiments based on the set *QAPLIB*, a well established data set in the literature, where instances of size n varying from 6 to 20 have been selected.

The *ILOG CPLEX 7.0 Concert Technology* has been used to implement the Adams and Johnson (RLT level-1) and the proposed reduced formulation. The experiments are carried out in a SUN BLADE 100 workstation with a 500 MHz processor and 1 Gb of RAM memory.

Table 1
Computational results for the data set of QAPLIB

Instance		Compact RLT level1			RLT level 1 [AJ94]			Integer
name	size	lp bound	time [s]	bound quality	lp bound	time [s]	bound quality	solution
nug5	5	50.0	0	1.0000	50.0	1	1.0000	50
tai5a	5	12,902.0	0	1.0000	12,902.0	1	1.0000	12,902
nug6	6	86.0	0	1.0000	86.0	1	1.0000	86
tai6a	6	29,432.0	1	1.0000	29,432.0	1	1.0000	29,432
nug7	7	148.0	1	1.0000	148.0	3	1.0000	148
tai7a	7	53,976.0	1	1.0000	53,976.0	1	1.0000	53,976
nug8	8	202.6	6	0.9469	203.5	17	0.9509	214
tai8a	8	77,502.0	5	1.0000	77,502.0	7	1.0000	77,502
tai9a	9	93,501.0	26	0.9882	93,501.0	37	0.9882	94,622
lipa10a	10	473.0	50	1.0000	473.0	50	1.0000	473
scr10	10	26,659.0	12	0.9902	26,873.1	269	0.9982	26,922
tai10a	10	131,053.0	86	0.9706	131,098.0	160	0.9709	135,028
tai10b	10	1,172,043.4	47	0.9901	1,176,140.0	248	0.9936	1,183,760
chr12a	12	9,552.0	2	1.0000	9,552.0	725	1.0000	9,552
chr12b	12	9,742.0	1	1.0000	9,742.0	508	1.0000	9,742
chr12c	12	10,927.3	1	0.9795	11,156.0	1,068	1.0000	11,156
had12	12	1,621.5	1,633	0.9816	1,621.5	2,533	0.9816	1,652
nug12	12	522.7	308	0.9043	522.9	6,597	0.9047	578
scr12	12	29,766.5	59	0.9477	29,827.3	4,555	0.9496	31,410
had14	14	2,666.1	12,044	0.9788	2,666.1	14,778	0.9788	2,724
chr15a	15	9,456.5	3	0.9556	9,513.0	30,146	0.9613	9,896
chr15c	15	9,504.0	2	1.0000	9,504.0	3,622	1.0000	9,504
nug15	15	1,040.3	7,895	0.9046	1,041.0	131,923	0.9052	1,150
nug20	20	2,180.4	157,424	0.8484	2,181.6	2,310,985	0.8489	2,570

Table 1 shows some interesting results. It is clear that the Compact RLT level-1 formulation produces gaps of same magnitude of the RLT level-1 [AJ94], as can be seen on figure 2, however at a much cheaper computational cost, as can be seen on figure 1.

A natural conclusion can be observed on table 2: the sparser the flow matrix of an instance, the easier it is of solving it when applying the flow elimination rule. We recall that the sparsity of a given instance is computed accounting only the entries having both b_{kl} and b_{lk} equal to zero. Hence, a full triangular superior matrix has zero sparsity.

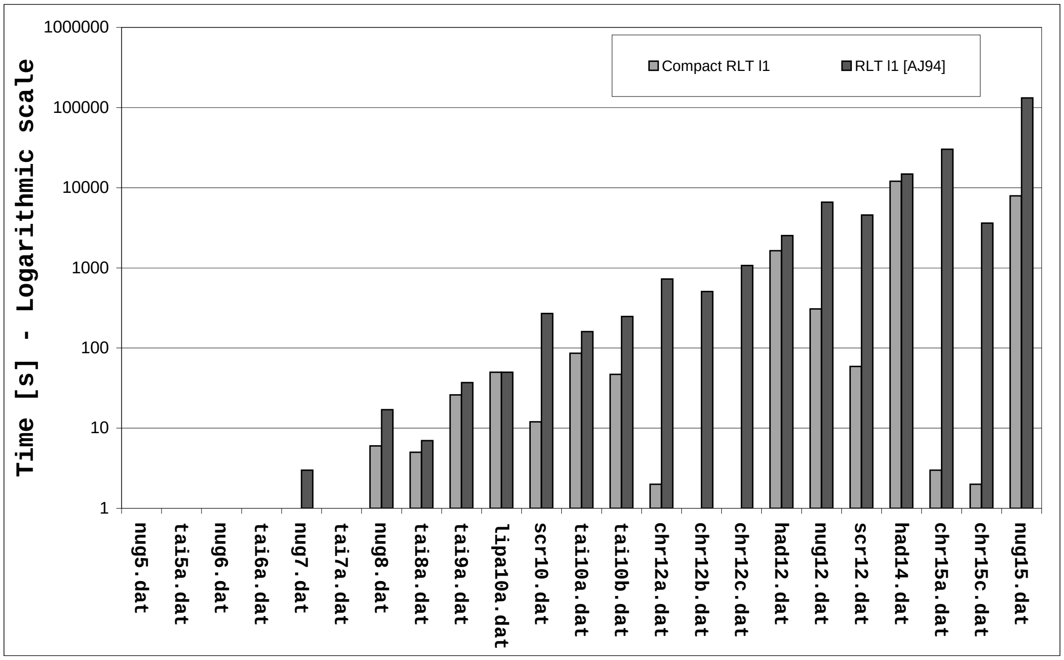

Fig. 1. Computational time comparison between both formulations.

Furthermore, we observe that sparsity becomes relatively more important as the problem size increases. The solution time improvement ratio is quite enormous in some test problems, as can be seen on table 2, fourth column. This is explained once any linear programming solver is very sensitive to the number of variables, so when the elimination of a *large* number of useless flow varibles is possible, one can expect a major impact on the computational times.

5 Conclusions

In this work we have presented a simple rule for improving the computational times of RLT level-1 bounds for the *Quadratic Assignment Problems* with sparse instances. This rule is based on the flow interpretation of the continuous variables of the cited formulation.

The flow elimination rule can obtain bounds that are hundreds or even thousands of times faster than the ones gotten by the traditional RLT level-1 formulation. Moreover, the proposed rule has not presented significant bound degeneracy for the tested instances.

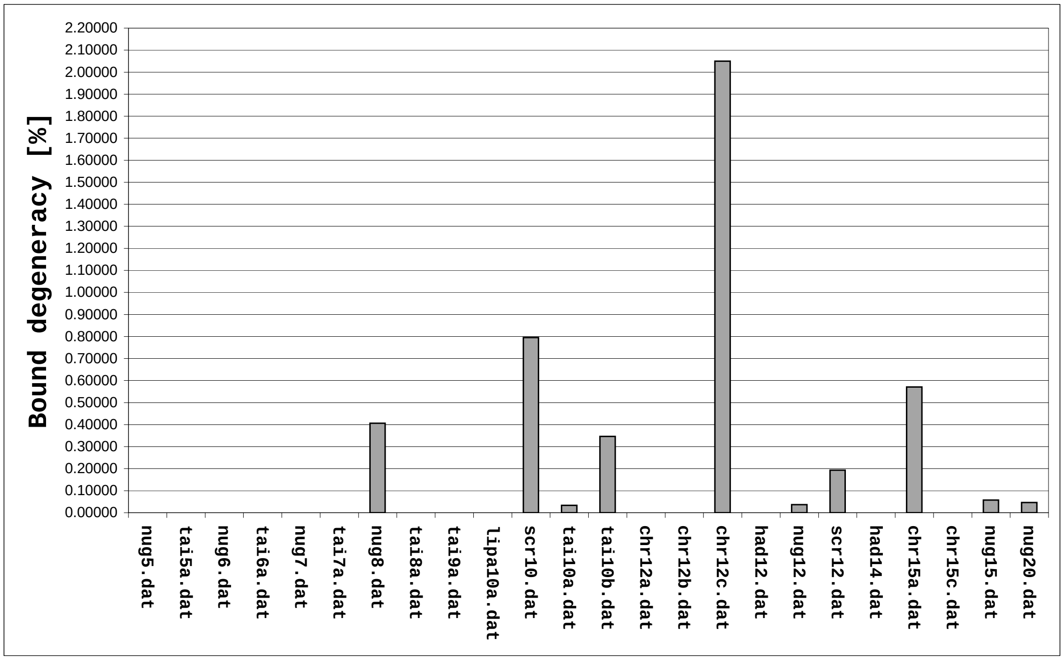

Fig. 2. Bound comparison between both formulations.

As a future research, it is possible to derive a rule to discard some constraints of the compact RLT level-1 formulation. The idea is to achieve a good balance between the bound strength and its cost.

References

[1] W. P. Adams and H. D. Sherali. Linearization strategies for a class of zero-one mixed integer programming problems. *Operations Research*, 38(2):217–226, March 1990.

[2] W.P. Adams and T. Johnson. Improved linear programming bounds for the quadratic assignment problem. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 16:43–77, 1994.

[3] K. M. Anstreicher. Recent advances in the solution of quadratic assignment problems. *Mathematical Programming*, 97:27–42, 2003.

[4] P. M. Hahn, W. L. Hightower, T. A. Johnson, M. Guignard-Spielberg, and C. Roucairol. A lower bound for the quadratic assignment problem based on a level-2 reformulation-linearization technique. Technical report, The University of Pennsylvania, USA, 2001.

Table 2
Solution time improvement ratio versus problem size and flow matrix sparsity.

Instance name	Instance size	Instance sparsity	Improvement ratio
tai5a.dat	5	0.100	1.00
nug5.dat	5	0.300	1.00
tai6a.dat	6	0.133	1.00
nug6.dat	6	0.333	1.00
tai7a.dat	7	0.047	1.00
nug7.dat	7	0.238	3.00
tai8a.dat	8	0.071	1.40
nug8.dat	8	0.357	2.83
tai9a.dat	9	0.055	1.42
lipa10a.dat	10	0.000	1.00
tai10a.dat	10	0.044	1.86
tai10b.dat	10	0.222	5.28
scr10.dat	10	0.511	22.42
had12.dat	12	0.000	1.55
nug12.dat	12	0.318	21.42
scr12.dat	12	0.575	77.20
chr12a.dat	12	0.833	362.50
chr12b.dat	12	0.833	508.00
chr12c.dat	12	0.833	1,068.00
had14.dat	14	0.000	1.23
nug15.dat	15	0.285	16.71
chr15a.dat	15	0.867	10,048.67
chr15c.dat	15	0.867	1,811.00
nug20.dat	20	0.257	14.68

[5] T.C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.

[6] K.G. Ramakrishnan, M.G.C. Resende, B. Ramachandran, and J.F. Penky. *Tight QAP Bounds Via Linear Programming.* World Scientific Publishing Co., Singapore, 1 edition, 2002.